(19)日本国特許庁(JP)　(12)**公開特許公報（A）**　(11)特許出願公開番号

特開平１０－８３２９３

(43)公開日　平成10年(1998)3月31日

(51)Int.Cl.⁶	識別記号	庁内整理番号	FI		技術表示箇所
Ｇ０６Ｆ　9/06	５４０		Ｇ０６Ｆ　9/06	５４０　Ｂ	
	４１０			４１０　Ｈ	

審査請求　未請求　請求項の数１５　ＯＬ　（全１０頁）

(21)出願番号　特願平9-76916

(22)出願日　平成9年(1997)3月28日

(31)優先権主張番号　08/625376
(32)優先日　1996年4月1日
(33)優先権主張国　米国（ＵＳ）

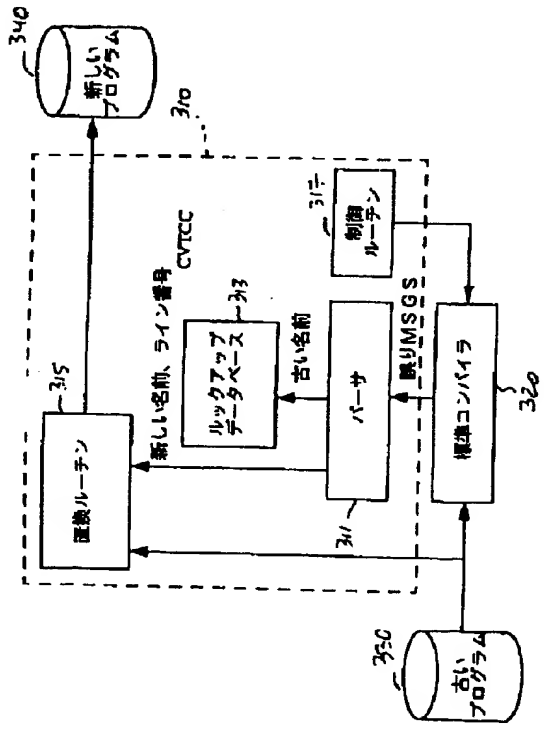
請求項3.

(71)出願人　594170738
サン　マイクロシステムズ　インコーポレイテッド
アメリカ合衆国　カリフォルニア州　94043　マウンテン　ヴィュー　ガルシア　アヴェニュー　2550
(72)発明者　ウォルター　ニールセン
アメリカ合衆国　カリフォルニア州　94404　フォスター　シティー　エッジウォーター　ブールヴァード　9696　アパートメント　704
(74)代理人　弁理士　中村　稔　（外6名）

最終頁に続く

(54)【発明の名称】誤り訂正コンパイラ

(57)【要約】
【課題】　ソフトウェア変換を自動化することができるツールを備えたシステム及び方法を提供する。
【解決手段】　一つのオペレーティング環境から別のオペレーティング環境へのソフトウェア・プログラムの自動化された変換を容易にするソフトウェア変換ツール。より特定的には、コンピュータ・プログラムは、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンパイルされる。結果として、コンパイラは、誤りメッセージを生成し、かつ誤りメッセージに応じて、コンピュータ・プログラム内のソース・コードは、誤りをもたらしている条件を除去するために自動的に変更される。ソフトウェア変換ツールは、誤り訂正コンパイラ、即ち、誤りを検出しかつそれらをユーザに与えるだけの代わりに、それらがリコンパイレーション中に発生しないように実際に誤りを訂正することができるコンパイラを結果として形成すべく、標準コンパイラでありうる、コンパイラと共同で動作する。



1

【特許請求の範囲】

【請求項１】　第２の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第１のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピューターインプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備することを特徴とするコンピューターインプリメンティッド方法。

【請求項２】　第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備することを特徴とする請求項１に記載の方法。

【請求項３】　請求項２に記載の段階を複数回繰り返すことを具備する請求項２に記載の方法。

【請求項４】　異なるプログラム識別子がデータベースに含まれるような各未知識別子が、その異なるプログラム識別子によって置換されるまで請求項２に記載の段階を複数回繰り返すことを具備する請求項３に記載の方法。

【請求項５】　コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで請求項２に記載の段階を複数回繰り返すことを具備する請求項３に記載の方法。

【請求項６】　コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイリング・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備することを特徴とするコンピューターインプリメンティッド方法。

【請求項７】　コンピュータ・プログラムをコンパイルする前に、コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルする段階を更に具備することを特徴とする

2

請求項６に記載の方法。

【請求項８】　変更する段階は、：誤りメッセージに応じて、未知プログラム識別子に対するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換することを具備することを特徴とする請求項７に記載の方法。

【請求項９】　第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備することを特徴とする請求項８に記載の方法。

【請求項１０】　請求項９に記載の段階を複数回繰り返すことを具備する請求項９に記載の方法。

【請求項１１】　コンパイラが誤りメッセージを生成しなくなるまで請求項９に記載の段階を繰り返すことを具備する請求項１０に記載の方法。

【請求項１２】　コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで請求項９に記載の段階を繰り返すことを具備する請求項１０に記載の方法。

【請求項１３】　第１のソフトウェア環境に対して書かれたプログラムを第２の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第２の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プログラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に応答する置換コードを備えていることを特徴とするコンピュータ・ソフトウェア・システム。

【請求項１４】　それにより変換されたプログラムを生成すべくシステムを反復的に置換及びリコンパイルさせるコンパイラ、パーサ、データベース、及び置換コードに結合された制御コードを更に備えていることを特徴とする請求項１３に記載の装置。

【請求項１５】　その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラ

ム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えていることを特徴とするコンピュータ読取り可能コード。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、ソフトウェア開発ツールに関し、特にソフトウェア変換ツールに関する。

【０００２】

【従来の技術】ソフトウェア開発は、時間が掛かりかつ面倒な処理である。先に書かれたソフトウェアを保守しかつ再加工することは、少なくとも同様に面倒である。幾度も、一つのソフトウェア環境において走るべく書かれたプログラムを別の異なるソフトウェア環境において走るフォームに変換するための必要性が生じる。そのような変換は、古いシステムを参照するプログラムにおける全ての言語構成要素を見出しかつ新しいシステムを参照する言語構成要素でそれらを置換することを必要とする。プログラムが大きく、数十万または百万のコードのラインを含んでいるならば、手動でそのような変換を実行することは、膨大なタスクである。明らかに、そのような変換を実行する機械援助型方法を工夫することが望ましい。

【０００３】

【発明が解決しようとする課題】残念ながら、特定テキスト・ストリングの全ての発生が異なる特定テキスト・ストリングで置換されるような、簡単な“探索及び置換”戦略は、有用な結果を供給するにはあまりにも簡単すぎる。自然言語におけるようにコンピュータ言語において、特定言語構成要素の意味は、かなりコンテキスト－依存性である。時々、例えば、ネームのローカル・バージョンが現れる。ネームのローカル・バージョンを汎用の新しいネームで置換することは、走らないかまたは正しく走らないプログラムを結果として生ずる。従って、上記した型のソフトウェア変換を、可能な限り、自動化するソフトウェア・ツールに対する必要性が存在する。

【０００４】本発明の目的は、上記従来の技術の必要性に応じて、ソフトウェア変換を自動化することができるツールを備えたシステム及び方法を提供することである。

【０００５】

【課題を解決するための手段】本発明の上記目的は、第２の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第１のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピューターインプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備するコンピューターインプリメンティッド方法によって達成される。

【０００６】本発明の方法では、第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備するようにしてもよい。本発明の方法では、上記記載の段階を複数回繰り返すことを具備するようにしてもよい本発明の方法では、異なるプログラム識別子がデータベースに含まれるような各未知識別子が、その異なるプログラム識別子によって置換されるまで上記記載の段階を複数回繰り返すことを具備するようにしてもよい。本発明の方法では、コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで上記記載の段階を複数回繰り返すことを具備するようにしてもよい。

【０００７】また、本発明の上記目的は、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイリング・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備するコンピューターインプリメンティッド方法によって達成される。本発明の方法では、コンピュータ・プログラムをコンパイルする前に、コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルする段階を更に具備するようにしてもよい。

【０００８】本発明の方法では、変更する段階は、：誤りメッセージに応じて、未知プログラム識別子に対するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換することを具備するようにしてもよい。本発明の方法では、第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備するようにしてもよい。本発明の方法では、上記記載の段階を複数回

繰り返すことを具備するようにしてもよい。本発明の方法では、コンパイラが誤りメッセージを生成しなくなるまで上記記載の段階を繰り返すことを具備するようにしてもよい。

【０００９】本発明の方法では、コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで上記記載の段階を繰り返すことを具備するようにしてもよい。更に、本発明の上記目的は、第１のソフトウェア環境に対して書かれたプログラムを第２の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第２の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プログラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に応答する置換コードを備えているコンピュータ・ソフトウェア・システムによって達成される。

【００１０】本発明のシステムでは、それにより変換されたプログラムを生成すべくシステムを反復的に置換及びリコンパイルさせるコンパイラ、パーサ、データベース、及び置換コードに結合された制御コードを更に備えてもよい。本発明の上記目的は、その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えているコンピュータ読取り可能コードによって達成される。

【００１１】

【作用】本発明は、一般的に言えば、一つのオペレーティング環境から別のオペレーティング環境へのソフトウェア・プログラムの自動化された変換を容易にするソフトウェア変換ツールを提供する。本発明の一実施例によれば、コンピュータ・プログラムは、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンパイルされる。結果として、コンパイラは、誤りメッセージを生成する。誤りメッセージに応じて、コンピュータ・プログラム内のソース・コードは、誤りをもたらす条件を除去するために自動的に変更される。ソフトウェア変換ツールは、誤り訂正コンパイラ、即ち、誤りを検出しかつユーザにそれらを与えるだけの代わりに、それらがリコンパイレーション中に発生しないように誤りを実際に訂正することができるコンパイラを結果として形成すべく、標準コンパイラでありうる、コンパイラと共同して動作する。

【００１２】本発明は、添付した図面に関する以下の説明から更に理解されうる。

【００１３】

【実施例】誤り訂正コンパイラが用いられうるコンピュータ・システムがまず説明され、より特定的に誤り訂正コンパイラそれ自体の説明が続く。図１をまず参照すると、誤り訂正コンパイラは、モニタ１１０、キーボード１２０及びマウス１３０を有している、図示された型のコンピュータ・ワークステーションで用いられうる。マウス・カーソル１４０は、モニタ画面上に現れかつマウスが動かされるように動く。図１のコンピュータ・ワークステーションの内部ハードウェアは、図２に示すような簡略化されたブロック図の形で表されうる。コンピュータ・システムは、中央処理装置２０２、システム・メモリ２０４、ディスプレイ２０６、プリンタ２０８、キーボード２１０、ポインティング・デバイス２１２、ディスク記憶サブシステム２１４、Ｉ／Ｏコントローラ２１６、及びシステム・バスのような相互接続手段２１８を有する。

【００１４】本発明のハードウェア環境を説明したので、誤り訂正コンパイラのソフトウェア・コンポーネント及びそれらの相関関係をより特定的に図３を参照してここで説明する。誤り訂正コンパイラは、実際にはソフトウェア変換プログラムと共同で動作している標準コンパイラである。一実施例では、ソフトウェア変換プログラムは、一つのアプリケーション・インターフェイス（例えば、NextStep）とコンパチブルなオブジェクティブＣプログラムを別のアプリケーション・インターフェイス（例えば、OpenStep）とコンパチブルなオブジェクティブＣプログラムに変換するために用いられ、かつオブジェクティブＣの一つのバージョンから別のバージョンにコードを変換するプログラムによって実行される機能を示している、“cvtcc”と名前が付けられる。

【００１５】図３を参照すると、変換プログラム３１０は、４つの主コンポーネント、単純構文解析系（シンプル・パーサ）３１１、データベース３１３、置換(substitution)プログラム３１５及び制御プログラム３１７を有する。変換プログラム３１０は、古いソフトウェア環境を想定する古いプログラム３３０を、新しいソフトウェア環境を想定する新しいプログラム３４０の中に変換

すべく標準コンパイラ３２０と共同で動作する。パーサ３１１及び置換プログラム３１５は、アクセス・プログラムとして一緒に機能する。データベース３１３及びアクセス・プログラムは、古い言語構成要素を新しい言語構成要素の中にマップする。特に、パーサ３１１は、コンパイラ２０から誤りメッセージを受け取りかつ古い資源名及びファイル名及び古い資源名が現れるライン番号を誤りメッセージから抽出する。古い名前は、データベース３１３における対応している新しい名前を調べるために用いられる。新しい名前、ファイル名及びライン番号は、次いで、古い名前に対する新しい名前の実際の置換を実行する、置換プログラム３１５に供給される。データベースは、古いソフトウェア環境及び新しいソフトウェア環境の両方に慣れている有能なプログラマによって手動で構築される。

【００１６】古いプログラムを新しいプログラムに変換するために、変換プログラム３１０は、制御プログラム３１７によって繰り返し走らされる。まず、ユーザは、古いプログラムをコンパイルするために、古いシステム環境の代わりに新しいシステム環境にコンパイルすべく設計または構成された、コンパイラを呼出す。結果として、古いシステム言語構成要素の使用で誤りが生成される。制御プログラムの制御下で、誤り訂正コンパイラ（即ち、コンパイラ３２０と共同する変換プログラム３１０）は、一つ以上の古いシステム構成要素を変換データベースにおいて見出される対応している新しい構成要素で繰り返し置換しかつリコンパイルする。それゆえに、誤り訂正コンパイラは、それ以上の誤りをフィックスすることができなくなるまで合成誤りをフィックスするような、実際のコンパイラを呼出し続ける。

【００１７】より特定的に図４を参照すると、誤り訂正コンパイラが走り始めるとき、フラグfixedSome （フィックスド・サム）は、false （フォールス）に設定される（ステップ４０１）。レギュラ・コンパイラは、次いで、変換されるべきプログラムをコンパイルすべく呼出され、かつコンパイラからの誤りリスティングが捕捉される（ステップ４０３）。次いで、最初の誤りメッセージがパースされる（ステップ４０５）。誤りが認識される型のものではないならば（ステップ４０７）、そして次の誤りメッセージが存在するならば（ステップ４０９）、それは、パースされる、等。誤りが認識される型、即ち、“identifier not found” （識別子が見出されなかった）メッセージのような、誤り訂正コンパイラがフィックスすることが可能でありうる型のものであれば（ステップ４０7）、古いプログラム的構成要素の名前は、古い名前への参照の位置（ファイル、ライン番号）と一緒に、誤りメッセージから抽出される（ステップ４１１）。古い名前は、次いで、データベースにおいて調べられる（ステップ４１３）。古い名前がデータベースにおいて見出されなければ（ステップ４１５）、次の誤りメッセージ（あれば）が処理される。古い名前がデータベースにおいて見出されるならば（ステップ４１５）、古い名前は、データベースにおいて見出される新しい名前で置換され（ステップ４１７）、その後フラグfixedSome は、真に設定される（ステップ４１９）。

【００１８】上記オペレーションのシーケンスは、次の誤りメッセージが存在しなくなるまで誤りメッセージの全てに対して実行される（ステップ４０９）。次いで、fixedSome が真であれば（ステップ４２１）、処理は、fixedSome をfalse にリセットし（ステップ４０１）かつ変更されたプログラムをリコンパイルする（ステップ４０３）ことによって、始めから繰り替えされる。オペレーションは、誤り訂正コンパイラがそれ以上の誤りをフィックスすることができなくなるまで、即ち、ステップ４２１でfixedSome がfalse に留まるまでこのファッションで進行する。次いで誤り訂正コンパイラは、停止する。誤り訂正コンパイラがフィックスすることができなかった誤りは、次いで、手動でフィックスされうる。

【００１９】誤り訂正コンパイラは、次に示す添付資料からさらに理解されうる。表１に示す添付資料Ａは、サンプル・データベースのリスティングである。表２に示す添付資料Ｂは、誤り訂正コンパイラの使用中に生成された画面プリントアウトである。

【００２０】

【表１】

```
appkit/Menu.h
 AppKit/NSMenu.h
@interface Menu : Panel
 @interface NSMenu : NSPanel
+ setMenuZone:(NSZone *)aZone;
 + (void)setMenuZone:(NSZone *)aZone;
+ (NSZone *)menuZone;
 + (NSZone *)menuZone;
+ init;

- initTitle:(const char *)aTitle;
 - (id)initWithTitle:(NSString *)aTitle;
 // if aTitle is NULL, change it to ""

 - (id)insertItemWithTitle:(NSString *)aString action:(SEL)aSelector keyEquivalent:(NSUI
- addItem:(const char *)aString action:(SEL)aSelector keyEquivalent:(unsigned short)chArCo
 - (id)addItemWithTitle:(NSString *)aString action:(SEL)aSelector keyEquivalent:(NSString
- setSubmenu:aMenu forItem:aCell;
 - (NSMenuCell *)setSubmenu:(NSMenu *)aMenu forItem:(NSMenuCell *)aCell;

 - (NSMenu *)supermenu;

 - (NSMenu *)attachedMenu;

 - (BOOL)isAttached;

 - (BOOL)isTornOff;
- itemList;
 - (NSMatrix *)itemMatrix;
- setItemList:aMatrix;
 - (void)setItemMatrix:(NSMatrix *)aMatrix;
- display;

- sizeToFit;
 - (void)sizeToFit;

- windowMoved:(NXEvent *)theEvent;

- close;

- update;

- setAutoupdate:(BOOL)flag;
 - (void)setAutoenableItems:(BOOL)flag;

 - (BOOL)autoenablesItems;
- findCellWithTag:(int)aTag;
 - (id)cellWithTag:(int)aTag;
- getLocation:(NXPoint *)theLocation forSubmenu:aSubmenu;
 - (NSPoint)locationForSubmenu:(NSMenu *)aSubmenu;
- mouseDown:(NXEvent *)theEvent;

- rightMouseDown:(NXEvent *)theEvent;

+ new;
- newTitle:(const char *)aTitle;

@end
 @end
@interface Menu(SubmenuDummyAction)
 @interface NSMenu(NSSubmenuDummyAction)
- submenuAction:sender;
 - (void)submenuAction:(id)sender;
@end
 @end

@interface NSObject(NSMenuActionResponder)

- (BOOL)validateCell:(id)aCell;

@end
```

【００２１】

【表２】

```
1  >>:271 more ~/all.m
2  #import <appkit/appkit.h>
3
4  void gus() {
5    Menu *fred;
6    [fred initTitle: "abc"];
7  }
8  >>:272 cc ~/all.m -c -I/net/dee/export/odp/dev8/jjb/nsm/src/include \
9          -I/usr/openwin/include/X11 -I/usr/openwin/include
10 "/home/jjb/all.m", line 2: Error: Could not open include file <appkit/appkit.h>.
11 "/home/jjb/all.m", line 5: Error: Menu is not defined.
12 "/home/jjb/all.m", line 5: Error: fred is not defined.
13 "/home/jjb/all.m", line 6: Error: fred is not defined.
14 4 Error(s) detected.
15 >>:273 bin/ercc ~/all.m -c -I/net/dee/export/odp/dev8/jjb/nsm/src/include \
16        -I/usr/openwin/include/X11 -I/usr/openwin/include
17 Using default fileName: /home/jjb/myaux/src/ConversionTools/appkitdb
18 *** Compiling ***
19 *** Attempting to fix errors ***
20 *** Fixed these msgs ***
21 "/home/jjb/all.m", line 2: Error: Could not open include file <appkit/appkit.h>.
22   — from> source= 'appkit/appkit.h'
23   —   to> source= ' AppKit/AppKit.h'
24 *** Fixed some missing include file errors; recompiling ***
25 *** Attempting to fix errors ***
26 *** Fixed these msgs ***
27 "/home/jjb/all.m", line 5: Error: Menu is not defined.
28   — from> file=appkit/Menu.h, source= '@interface Menu : Panel'
29   —   to> file=AppKit/NSMenu.h, source= ' @interface NSMenu : NSPanel'
30
31 *** Fixed some errors, recompiling ***
32 *** Attempting to fix errors ***
33 *** Fixed these msgs ***
34 "/home/jjb/all.m", line 6: Warning: Cannot find instance method initTitle: \
35                       for class NSMenu:NSPanel:NSWindow:NSResponder:NSObject.
36   — from> file=appkit/Menu.h, source= '- initTitle:(const char *)aTitle;'
37   —   to> file=AppKit/NSMenu.h, source= ' - (id)initWithTitle:(NSString *)aTitle;'
38      /* if aTitle is NULL, change it to @"" */
39
40 *** Fixed some errors, recompiling ***
41 *** Attempting to fix errors ***
42
43 *** Fixed all we can in this file ***
44
45
46 >>:274 more ~/all.m
47
48 #import <AppKit/AppKit.h>
49
50
51 void gus() {
52   NSMenu *fred;
53   [fred initWithTitle: "abc"];
54 }
```

添付資料Ａを参照すると、説明された実施例における、データベースは、非常に簡単な構成である。最初のテキスト・カラムで始まっているエントリは、置換されるべき古い名前である。置換エントリがデータベースにおいて見出されるならば、それは、第２のテキスト・カラムで始まっている、直後のラインに入力される。全ての古い名前が対応している新しい名前をもっているわけではなく、その逆もそうである。記号“#”、“@”、“+”、“－”は、オブジェクティブＣにおいて認識された意味を有する。添付資料Ｂを参照すると、誤り訂正コンパイラのオペレーションの具体的な例を供給するために、SparcStation（スパークステーション）・コンピュータで走っておりかつSunOS （サンオーエス）オペレーティング・システム下で走っている誤り訂正コンパイラのバージョンは、ライン１～７の例示プログラムに呼出された。ライン８から始まって、コンパイラが走らせ、ライン１１～１４にリストされた４つの誤りメッセージを結果として生ずる。ライン１６から始まって、変換プログラムは、コンパイラによって生成された誤りメッセージ・ファイルを用いて走らさせる。変換プログラムは、組込みファイル誤りをフィックスすることをまず試みる。そこでライン２１～２４において、変換プログラムは、それが、データベースにおいて特定されたように組込みファイルの名前を変更したことを報告する。コンパイラは、次いで、変更されたプログラムをリコンパイルすべく再び呼出される（ライン２６）。更なる誤りメッセージに報告された誤りは、次いで、フィックスされる。ライン４４では、誤り訂正コンパイラは、可能な限り全ての誤りをフィックスしたことを報告する。合成変更プログラムは、ライン４８～５４にリストされる。オリジナル・プログラムの“Menu”は、“NSMenu”に変更され、かつ“initTitle ”は、“initWithTitle ”に変更されたということに注目する。

【００２２】本発明は、その精神または重要な特徴から逸脱することなく他の特定のフォームで実施することができるということが当業者によって理解されるであろ

う。上述の記載は、従って全ての関して説明のためであり限定的でなものではない。本発明の範疇は、特許請求の範囲によって示され、かつその同等物の意味及び範囲内にある全ての変更は、その中に含まれることを意図する。

【００２３】

【発明の効果】本発明のコンピューターインプリメンティッド方法は、第２の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第１のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピューターインプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第２の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備するので、ソフトウェア変換を可能な限り自動化することができる。

【００２４】本発明のコンピューターインプリメンティッド方法は、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイリング・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備するので、ソフトウェア変換を可能な限り自動化することができる。本発明のコンピュータ・ソフトウェア・システムは、第１のソフトウェア環境に対して書かれたプログラムを第２の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第２の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プログラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に応答する置換コードを備えているので、ソフトウェア変換を可能な限り自動化することができる。

【００２５】本発明のコンピュータ読取り可能コードは、その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えているので、ソフトウェア変換を可能な限り自動化することができる。

【図面の簡単な説明】

【図１】本発明が用いられうる通常のコンピュータ・ワークステーションの斜視図である；

【図２】図１のコンピュータ・ワークステーションの一部のブロック図である；

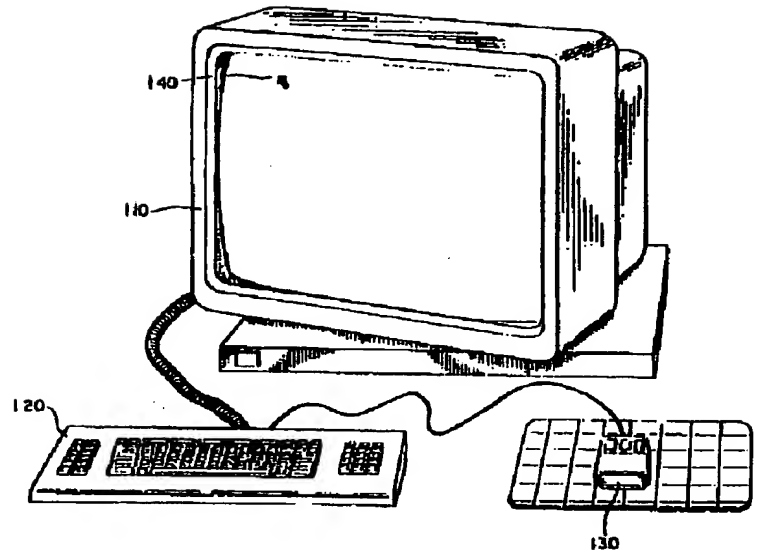
【図３】本発明の誤り訂正コンパイラのブロック図である；

【図４】図３の誤り訂正コンパイラのオペレーションを説明するフロー図である。

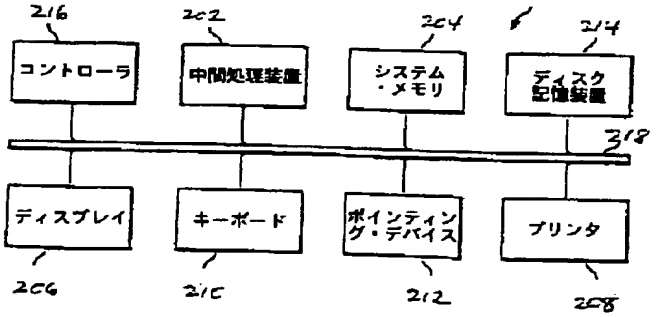
【符号の説明】

３１０　変換プログラム
３１１　単純構文解析系（シンプル・パーサ）
３１３　データベース
３１５　置換プログラム
３１７　制御プログラム
３２０　標準コンパイラ
３３０　古いプログラム
３４０　新しいプログラム

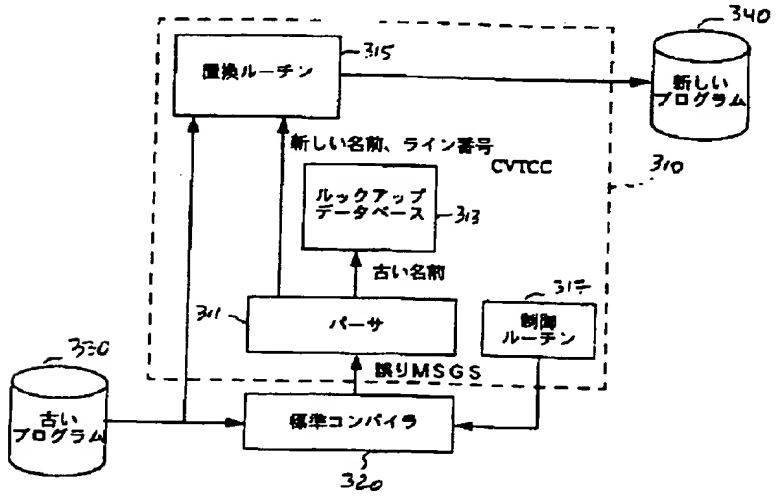
【図１】


140
110
120
130

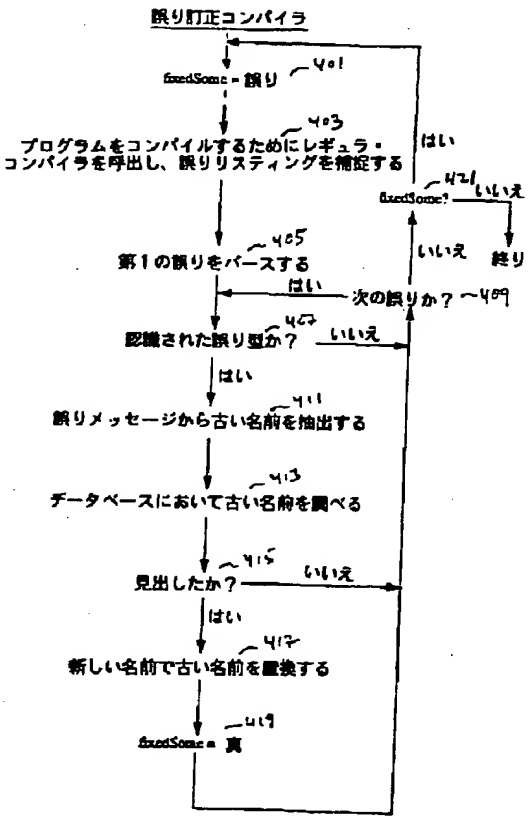
【図２】


216
コントローラ
202
中間処理装置
204
システム・メモリ
214
ディスク記憶装置
218
ディスプレイ
キーボード
ポインティング・デバイス
プリンタ
206
210
212
208

【図３】


340
置換ルーチン
315
新しいプログラム
新しい名前、ライン番号
CVTCC
310
ルックアップデータベース
313
古い名前
311
パーサ
317
制御ルーチン
誤りMSGS
350
古いプログラム
標準コンパイラ
320

【図４】


誤り訂正コンパイラ
401
fixedSome = 誤り
403
プログラムをコンパイルするためにレギュラ・コンパイラを呼出し、誤りリスティングを捕捉する
はい
421
fixedSome?
いいえ
終り
405
第１の誤りをパースする
いいえ
はい
次の誤りか？
409
認識された誤り型か？
407
いいえ
はい
411
誤りメッセージから古い名前を抽出する
413
データベースにおいて古い名前を調べる
415
見出したか？
いいえ
はい
417
新しい名前で古い名前を置換する
419
fixedSome = 真

フロントページの続き

(72)発明者　ジェームズ　ホームルンド
アメリカ合衆国　カリフォルニア州
94303　パロ　アルト　アーストワイルド
コート　73